The CADE ATP System Competition — CASC

Geoff Sutcliffe

■ *The CADE ATP System Competition (CASC) is an annual evaluation of fully automatic automated theorem-proving (ATP) systems for classical logic — the world championship for such systems. CASC provides a public evaluation of the relative capabilities of ATP systems, and aims to stimulate ATP research toward the development of more powerful ATP systems. Over the years CASC has been a catalyst for impressive improvements in ATP.*

The CADE ATP System Competition (CASC) is an annual evaluation of fully automatic automated theorem-proving (ATP) systems for classical logic — the world championship for such systems. CASC is held at the International Conference on Automated Deduction (CADE) or the International Joint Conference on Automated Reasoning (IJCAR, which replaces CADE on alternate years) each year. These conferences are the major forums for the presentation of new research in all aspects of automated deduction. The evaluation is in terms of the number of problems solved, the number of solutions output, and the average run time for problems solved in the context of a bounded number of eligible problems, chosen from the TPTP Problem Library (Sutcliffe 2009), and a CPU time limit for each solution attempt.

One purpose of CASC is to provide a public evaluation of the relative capabilities of ATP systems. Additionally, CASC aims to stimulate ATP research, motivate development and implementation of robust ATP systems that are useful and easily deployed in applications, provide an inspiring environment for personal interaction between ATP researchers, and expose ATP systems within and beyond the ATP community. Fulfillment of these objectives provides insight and stimulus for the development of more powerful ATP systems, leading to increased and more effective use.

The first CASC was held at CADE-13 in Nancy, France, in 1996, devised and organized by Christian Suttner and Geoff Sutcliffe.[1] The most recent CASC, held at CADE-25 in Berlin, Germany, in 2015, was the twentieth CASC in the series. Over the years CASC has been a catalyst for impressive improvements in ATP, stimulating both theoretical and implementation advances (Nieuwenhuis 2002). It has provided a forum at which empirically successful implementation efforts are acknowledged and applauded, and at the same time provides a focused meeting at which novice and experienced developers exchange ideas and techniques. The CASC web site provides access to all details of the individual competitions.[2]

CASC is run in divisions according to problem and system characteristics. Over the years, 12 divisions have existed (those marked with an asterisk (*) were the divisions for CASC-25): (1) THF:* typed higher-order form theorems (axioms with a provable conjecture). (2) THN:* typed higher-order form nontheorems (axioms with a countersatisfiable, that is, unprovable conjecture, and satisfiable axiom sets. (3) TFA:* typed first-order with arithmetic theorems (axioms with a provable conjecture). (4) TFN:* typed first-order with arithmetic nontheorems (axioms with a countersatisfiable conjecture, and satisfiable axiom sets). (5) FOF*: first-order form theorems (axioms with a provable conjecture). (6) FNT:* first-order form nontheorems (axioms with a countersatisfiable conjecture, and satisfiable axiom sets). (7) CNF: clause normal form theorems (unsatisfiable clause sets) that are not effectively propositional,[3] and not unit equality problems (see the UEQ division). (8) SAT: clause normal form nontheorems (satisfiable clause sets) that are not effectively propositional, and not unit equality problems (see the UEQ division). (9) EPR:* effectively propositional theorems and nontheorems (unsatisfiable and satisfiable clause sets). (10) UEQ: unit equality theorems (unsatisfiable clause sets) that are not effectively propositional. (11) SEM: FOF theorems based on a specified axiomatization of a specified semantic domain. (12) LTB:* first-order form theorems (axioms with a provable conjecture) from large theories, presented in batches with a shared time limit.

The different logics and syntactic characteristics of the problems in the various divisions provide different challenges for ATP systems. The tasks of proving theorems and showing unsatisfiability (which can be treated similarly) are quite distinct from establishing nonprovability and satisfiablility (which can also be treated similarly).

Problems for CASC are taken from the TPTP Problem Library. The TPTP version used for CASC is released after the competition, so that new problems have not been seen by the entrants. In some divisions the systems are ranked according to the number of problems solved with an acceptable proof/model output, and in some divisions the systems are ranked according to the number of problems solved but not necessarily accompanied by a proof or model (thus giving only an assurance of the existence of a proof/model). Ties are broken according to the average time over problems solved. Division winners are announced and prizes are awarded. In addition to the ranking criteria, three other measures are made and presented in the results: The state-of-the-art (SoTA) contribution quantifies the unique abilities of each system. For each problem solved by a system, its SoTA contribution for the problem is the inverse of the number of systems that solved the problem, and its overall SoTA contribution is the average SoTA contribution over the problems it solved. The efficiency measure is a combined measure that balances the time taken for each problem solved against the number of problems solved. It is the average of the inverses of the times for problems solved, This can be interpreted intuitively as the average of the solution rates for problems solved, multiplied by the fraction of problems solved. The core usage is the average of the ratios of CPU time to wall clock time used, over the problems solved. This measures the extent to which the systems take advantage of multiple cores.

CASC typically has 20 to 30 ATP systems entered. For each CASC the division winners of the previous CASC are automatically entered to provide benchmarks against which progress can be judged. Additionally, a fixed version (initially v3.2, later v3.3) of the well known Otter ATP system was entered in every CASC from 2002 to 2011, as a fixed point against which progress could be judged. By 2011 Otter was no longer competitive, and was replaced by Prover9 2009-11A in 2012. Over all 20 CASCs, so far 99 distinct ATP systems have been entered. Almost all the ATP systems have come from academia, partially due to the CASC requirement that all source code must be published on the CASC website. The most popular divisions have been the FOF, FNT, CNF, SAT, EPR, and UEQ divisions. Some systems have emerged as dominant in some of the divisions: Satallax in the THF division, Vampire in the FOF and CNF divisions, Paradox in the FNT and SAT divisions (with iProver now coming on strong), iProver in the EPR division, and Waldmeister in the UEQ division. The strengths of these systems stem from four main areas: solid theoretical foundations, significant implementation efforts (in terms of coding and data structures), extensive testing and tuning, and an understanding of how to optimize for CASC. For example, Vampire is founded on the theoretical principles of superposition, has a highly efficient implementation in C++

using code trees and advanced structures for representing logical data, is repeatedly tested and tuned on the TPTP problem library, and has special modes for the various divisions of CASC. Technical information about these systems, and the techniques they employ, can be found on the individual CASC web pages.

The design and organization of CASC have evolved over the years to a sophisticated state. Decisions made for CASC (alongside the TPTP) have had an influence on the directions of development in ATP. It is interesting to look back on some of the key decisions that have helped bring the competition to its current state.

CASC-13, 1996 — the first CASC — stimulated research toward robust, fully automatic systems that take only logical formulae as input. It increased the visibility of systems and developers, and rewarded implementation efforts. CASC-14, 1997, introduced the SAT division, stimulating the development of model-finding systems for CNF. CASC-15, 1998, introduced the FOF division, starting the slow demise of CNF to becoming just the assembly language of ATP. At CASC-16 in 1999, changes to the problem-selection process motivated the development of techniques for automatic tuning of ATP systems' search parameters. CASC-JC, 2001, introduced ranking based on proof output, starting the trend toward ATP systems that efficiently output proofs and models. CASC-JC also introduced the EPR division, stimulating the development of specialized techniques for this important subclass of problems. CASC-20, 2005, required systems to develop built-in equality reasoning, by removing the equality axioms from all TPTP problems. At CASC-J3, in 2006, the FOF division was promoted as the most important, stimulating development of ATP systems for full first-order logic. CASC-21, 2007, introduced the FNT division, further stimulating the development of model-finding systems. CASC-J4, 2008, introduced the LTB division, stimulating the development of techniques for automatically dealing with very large axiom sets. CASC-J5, 2010, introduced the THF division, stimulating development of ATP systems for higher-order logic. CASC-23, 2011, introduced the TFA division, stimulating development of ATP systems for full first-order logic with arithmetic. At CASC-J6, in 2012, Prover9 replaced Otter as the fixed-point in the FOF division, demonstrating the progress in ATP. CASC-24, 2013, removed the CNF division, confirming the demise of CNF. CASC-J7, 2014, required use of the SZS ontology, so the ATP systems unambiguously report what they have established about the problem. CASC-25, 2015, introduced the THN and TFN divisions, stimulating development of model finding for the THF and TFA logics.

Over the years TPTP and CASC have increasingly been used as a conduit for ATP users to provide samples of their problems to ATP system developers. Users' problems that are contributed to TPTP are eligible for use in CASC. The problems are then exposed to ATP system developers, who improve their systems' performances on the problems, in order to perform well in CASC. This completes a cycle that provides the users with more effective tools for solving their problems.

Notes

[1] Christian Suttner was a CASC organizer for the first 10 CASCs, and various other people have contributed to the running of selected CASC editions.

[2] www.tptp.org/CASC.

[3] Effectively propositional means that the problem is known to be reducible to a propositional problem, e.g., a CNF problem that has no functions with arity greater than zero.

References

Nieuwenhuis, R. 2002. Special Issue: The CADE ATP System Competition. *AI Communications* 15(2–3).

Sutcliffe, G. 2009. The TPTP Problem Library and Associated Infrastructure. The FOF and CNF Parts, v3.5.0. *Journal of Automated Reasoning* 43(4): 337–362. dx.doi.org/10.1007/s10817-009-9143-8

Geoff Sutcliffe is a professor and chair of the Department of Computer Science at the University of Miami. His research is in the area of automated reasoning, particularly in the evaluation and effective use of automated reasoning systems. His most prominent achievements are the first ever development of a heterogeneous parallel deduction system, leading to the development of the SSCPA automated reasoning system; the development and ongoing maintenance of the TPTP problem library, which is now the de facto standard for testing classical logic automated reasoning systems; the development and ongoing organization of the CADE ATP System Competition — the world championship for classical logic automated reasoning systems; and the specification of the TPTP language standards for automated reasoning tools.